

# Построение карт глубины. SGM

Фотограмметрия. Лекция 5



- ректификация стереопары
- Semi-Global Matching - **SGM**
- **tSGM**

Полярный Николай  
[polarnick239@gmail.com](mailto:polarnick239@gmail.com)

# Где мы сейчас?

- 1) Ключевые точки (SIFT)
- 2) Сопоставление ключевых точек (RANSAC, k-ratio test)
- 3) Определение взаимного расположения пар камер
- 4) Уточнение расположения камер и их калибровок (Bundle Adjustment)

Итого:

- Есть точная калибровка камеры и ракурсов фотографирования (intrinsics & extrinsics parameters)
- Есть точное **но разреженное (недетальное)** облако 3D точек

# Что дальше?

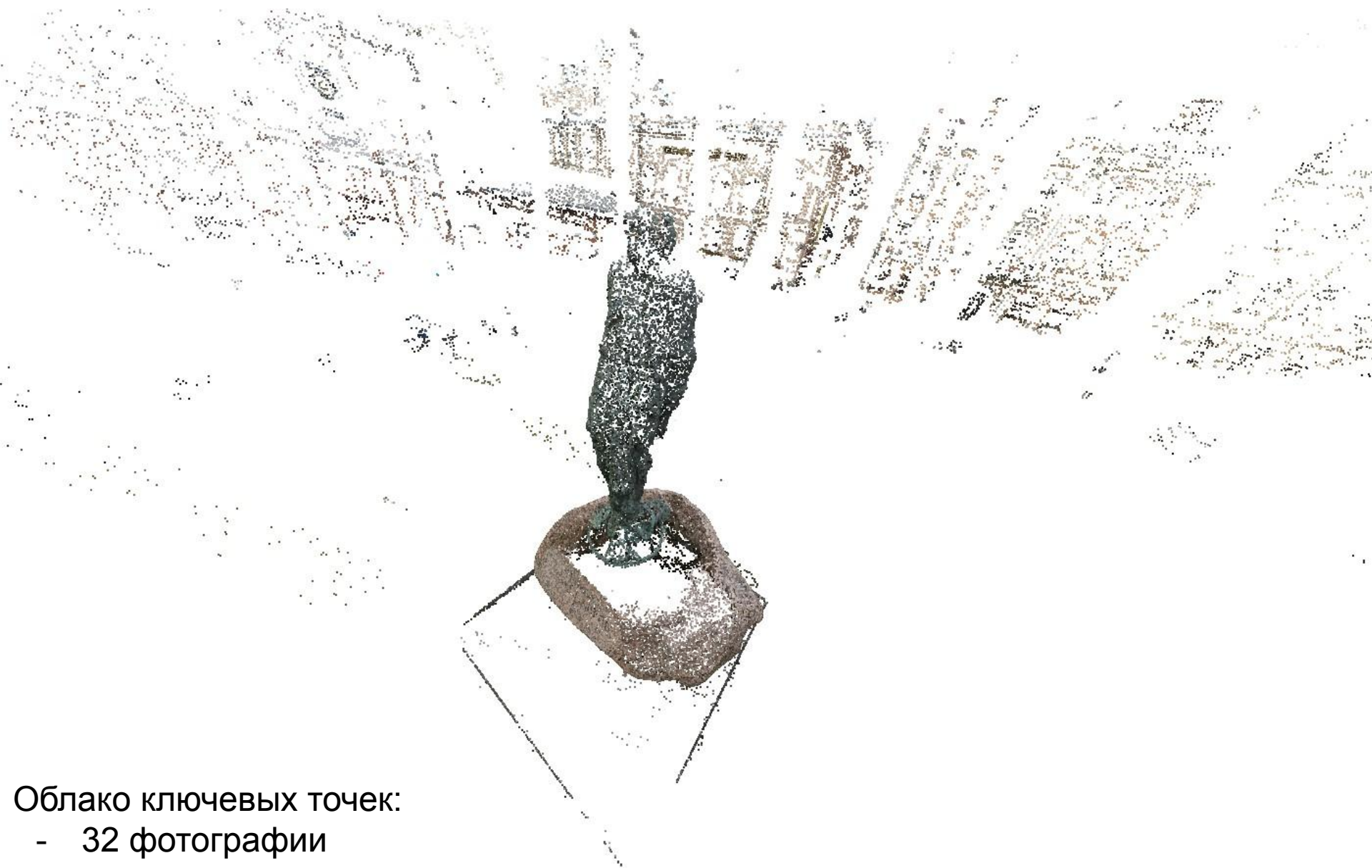
- 1) Ключевые точки (SIFT)
- 2) Сопоставление ключевых точек (RANSAC, k-ratio test)
- 3) Определение взаимного расположения пар камер
- 4) Уточнение расположения камер и их калибровок (Bundle Adjustment)

Итого:

- Есть точная калибровка камеры и ракурсов фотографирования (intrinsics & extrinsics parameters)
- Есть точное **но разреженное (недетальное)** облако 3D точек

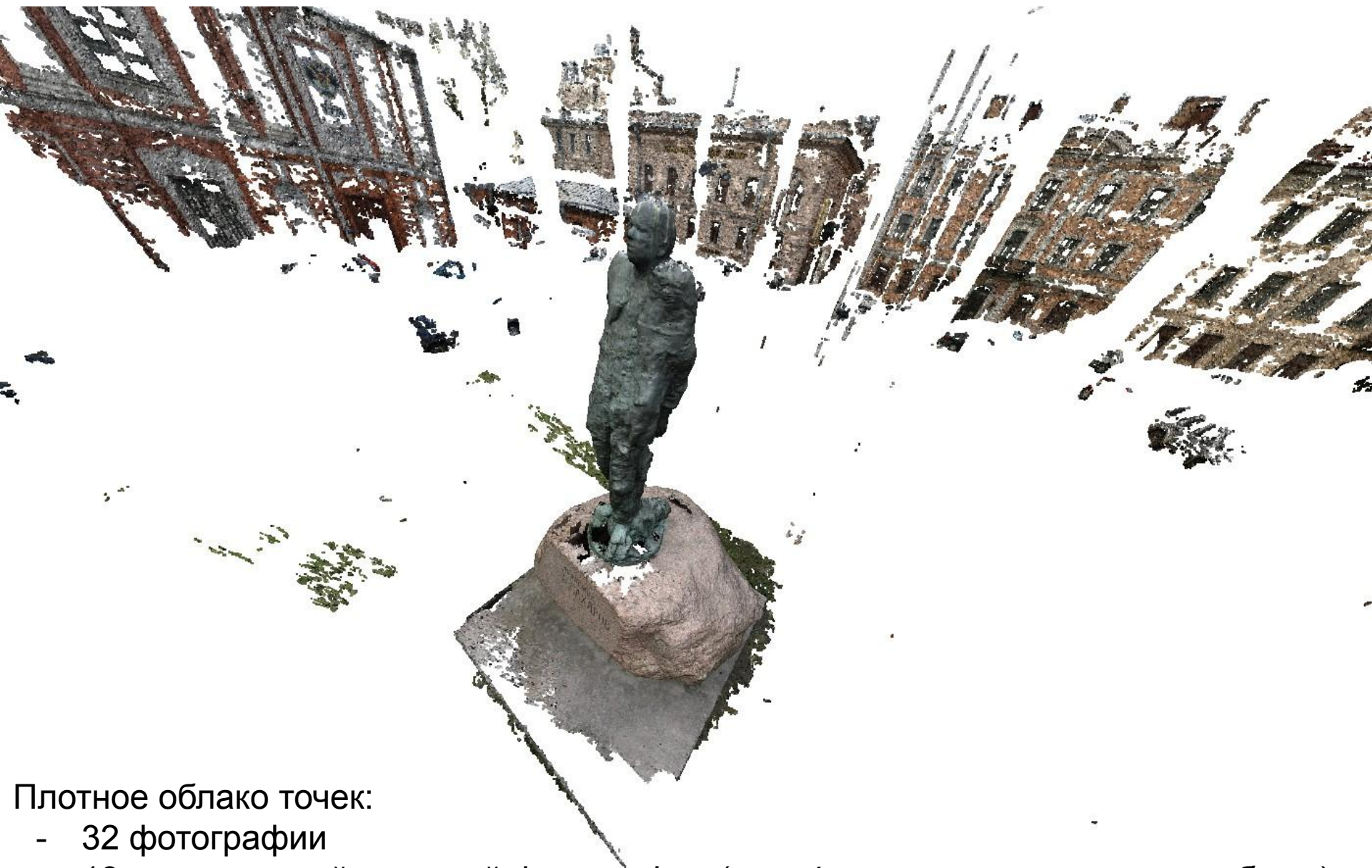
Хотим:

- Детальную геометрию, например облако точек:  
“**один пиксель - одна точка**”



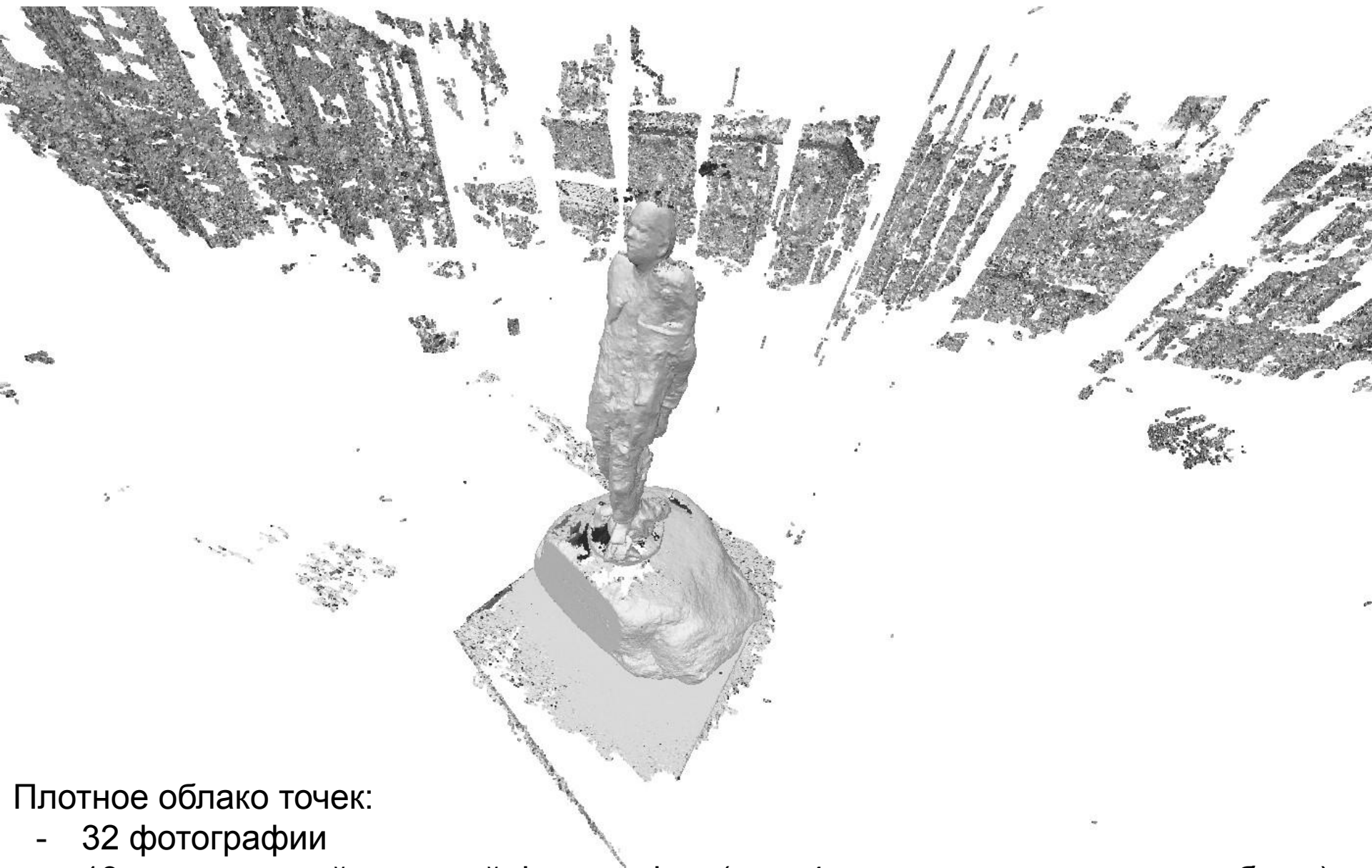
Облако ключевых точек:

- 32 фотографии
- ~8 тысяч ключевых точек на каждой фотографии
- **83 тысячи** 3D точек



Плотное облако точек:

- 32 фотографии
- 13 мегапикселей в каждой фотографии (и по 4 мегапикселя на карту глубины)
- **34 миллиона** 3D точек (построены из карт глубины)



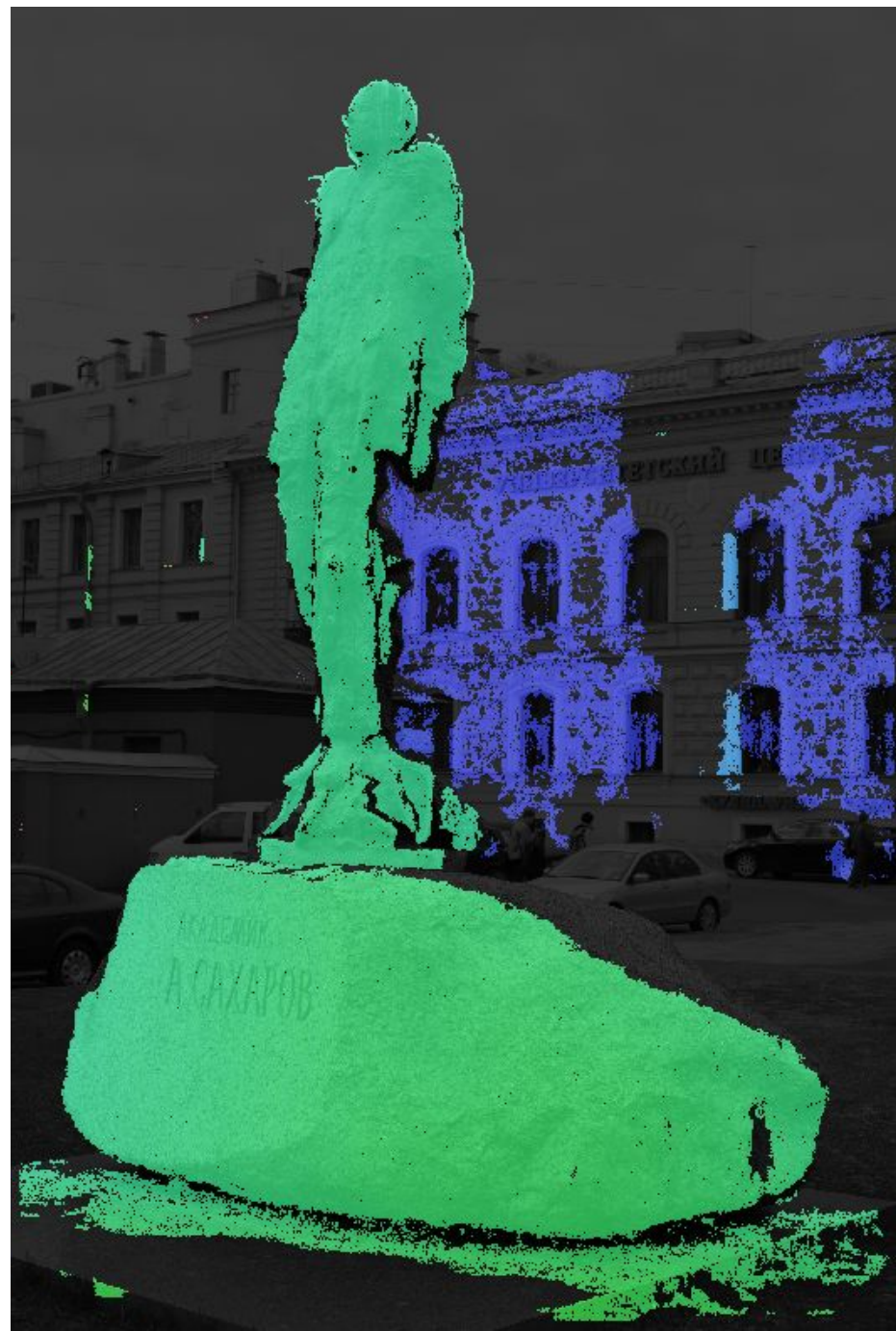
Плотное облако точек:

- 32 фотографии
- 13 мегапикселей в каждой фотографии (и по 4 мегапикселя на карту глубины)
- **34 миллиона** 3D точек (построены из карт глубины)











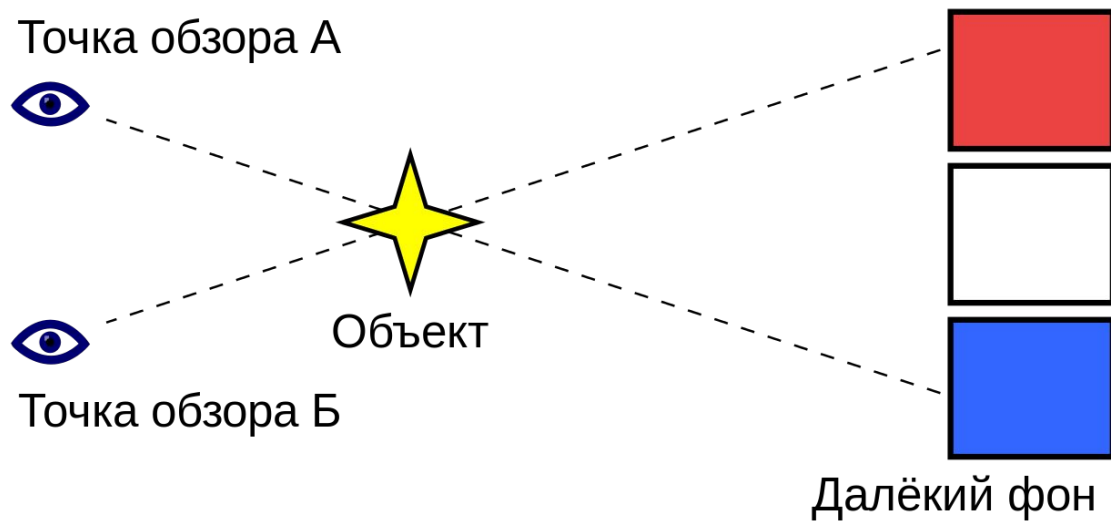
# План лекции

- 1) Ректификация стереопары (на базе эпиполярной геометрии)
- 2) Cost функции (оценка похожести патчей)
- 3) Semi-Global Matching (**SGM**)
- 4) Coarse-to-fine/иерархичный SGM (**SURE tSGM**)

# Стереопары

За счет чего человек “видит” расстояния до объектов и в целом сцену перед ним? (кроме жизненного опыта конечно)

За счет пары глаз (две картинки - **стереопара**) и смещения (**параллакса**):

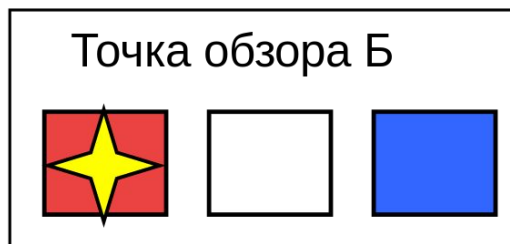
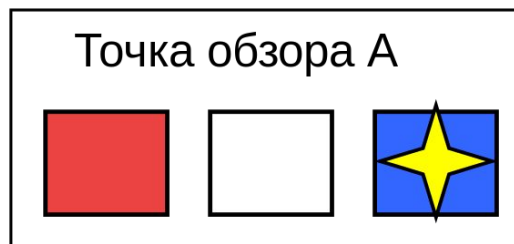


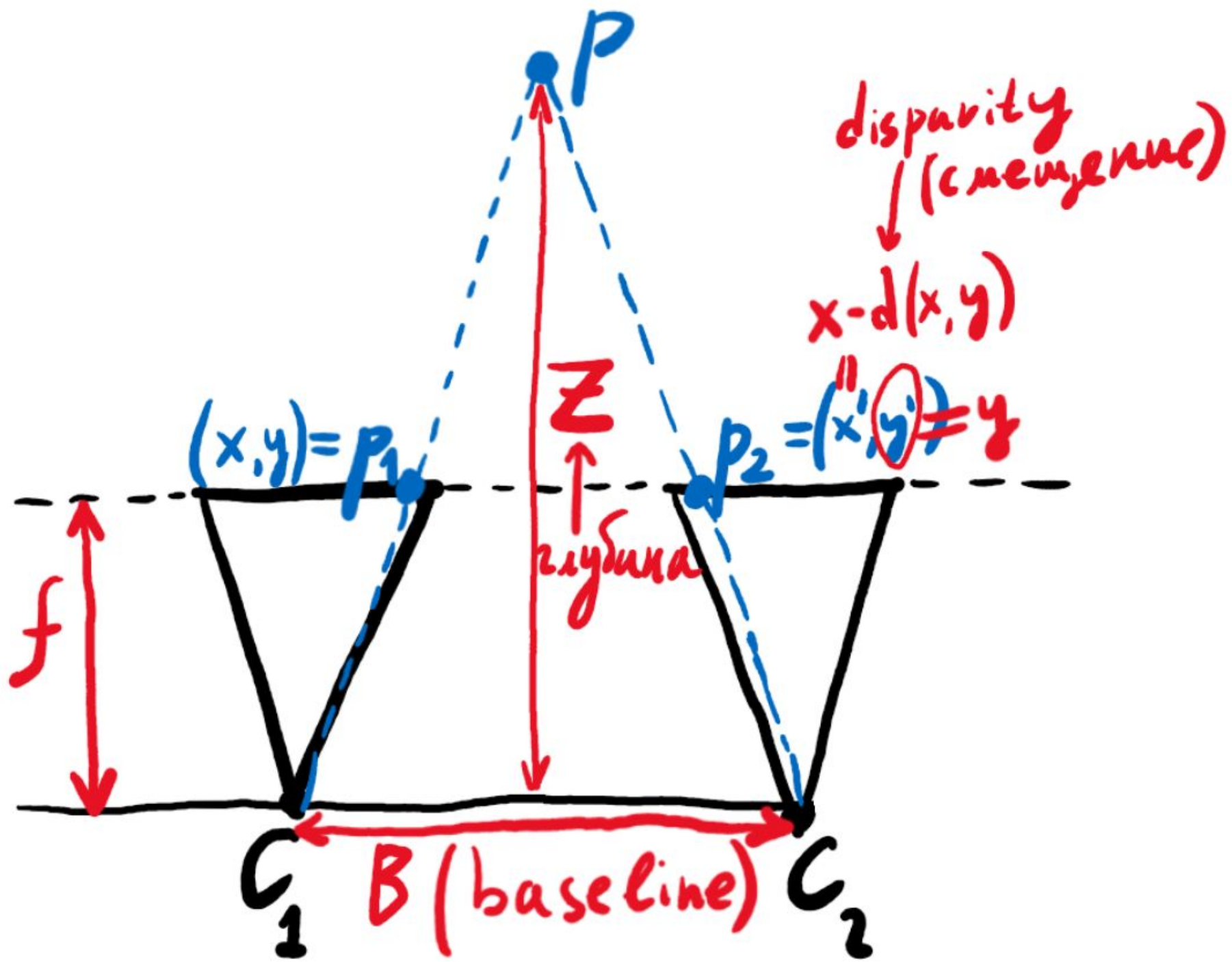
Почему смещение по горизонтали?

Что это за горизонтальная прямая? (моя любимая :D)

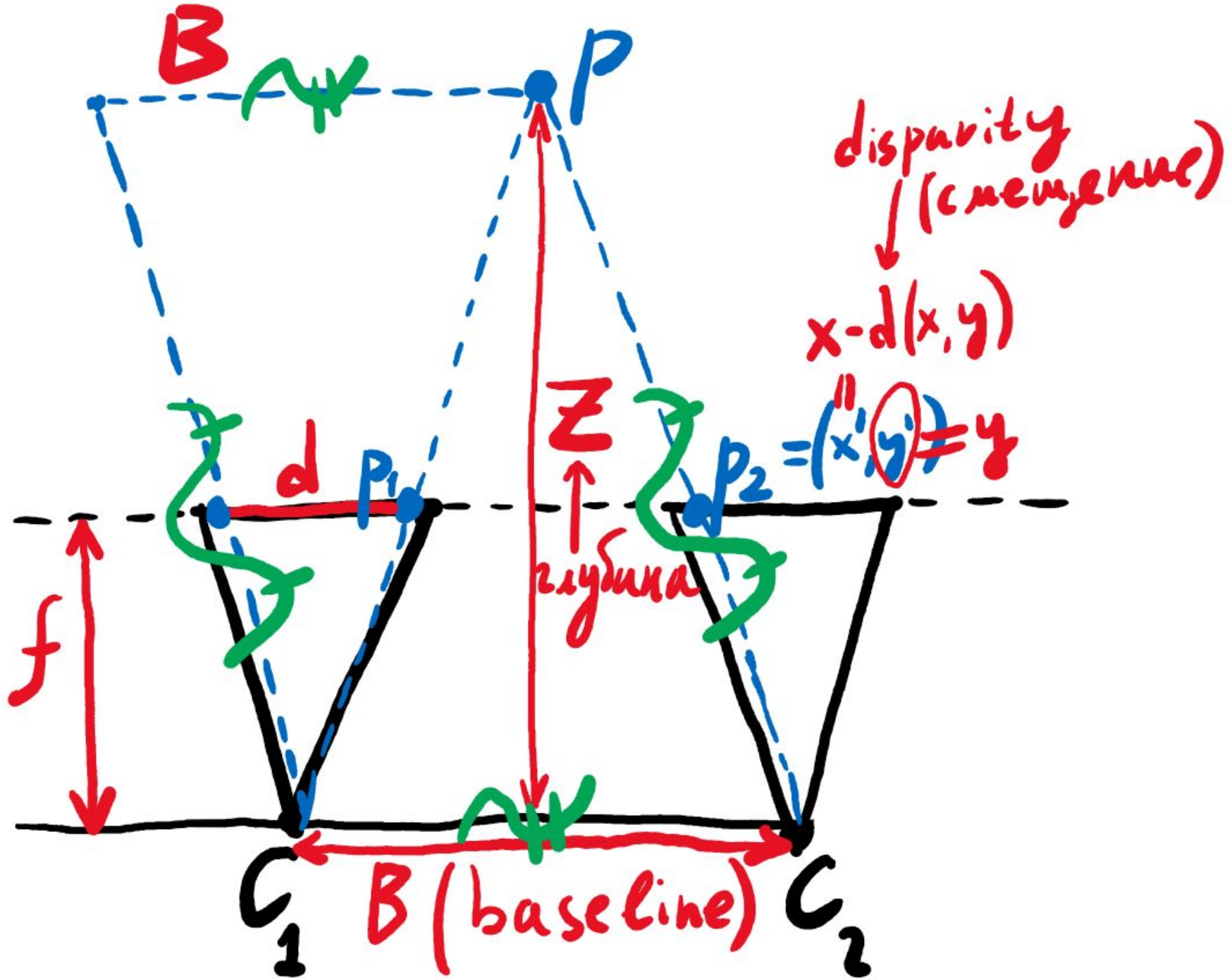
Какая зависимость расстояния до объекта от смещения в стереопаре?

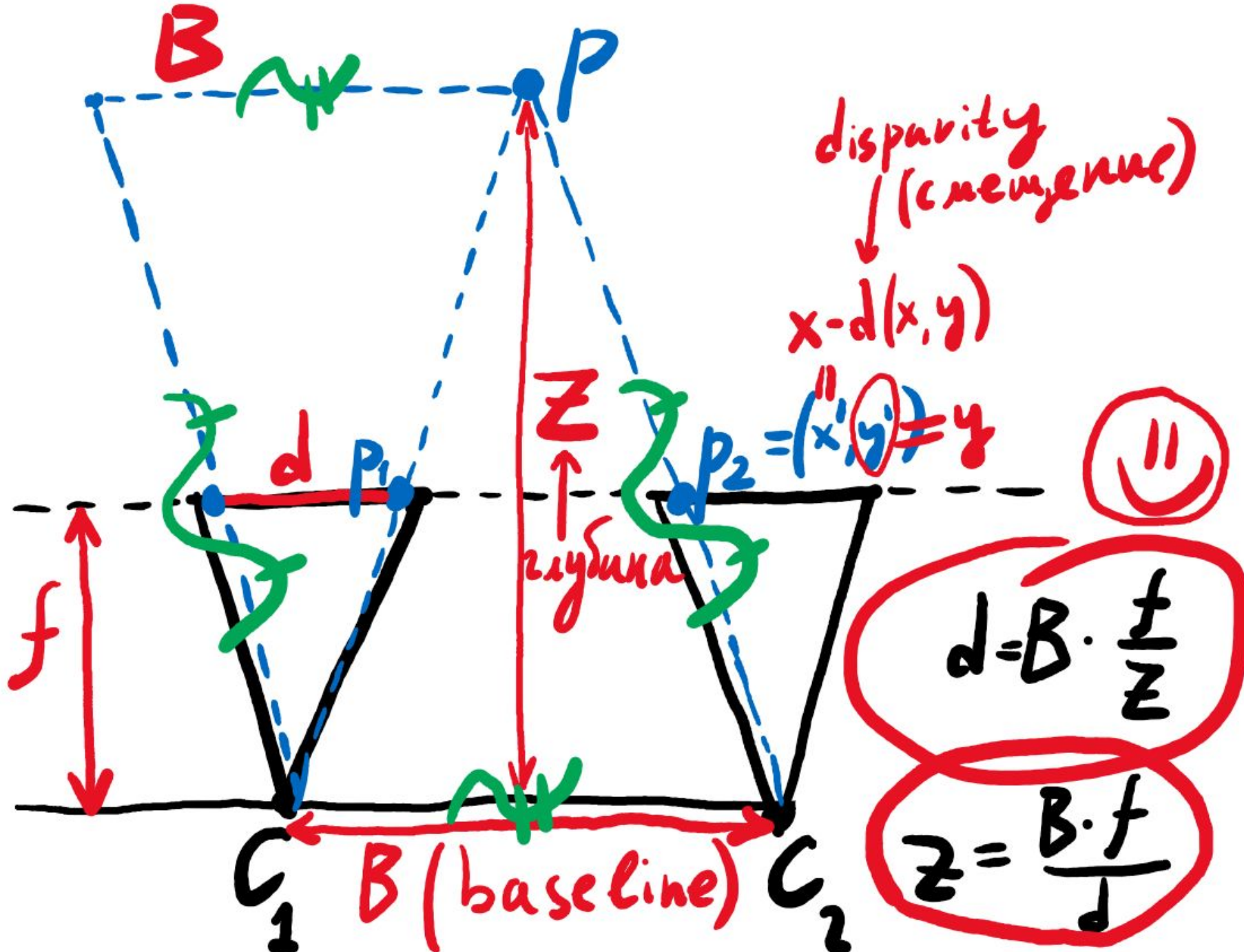
Как выглядит сопоставление для произвольной пары фотографий?







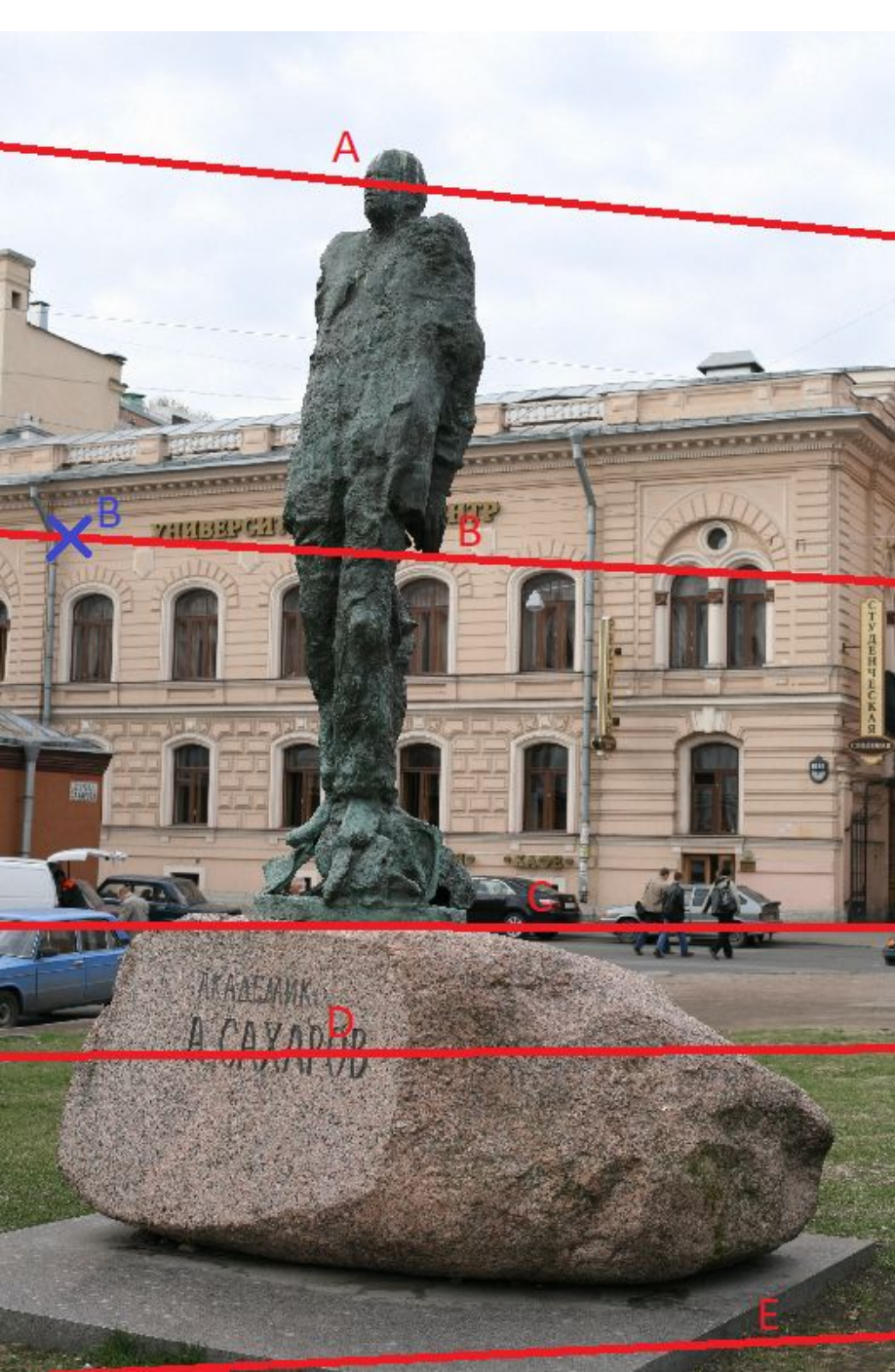




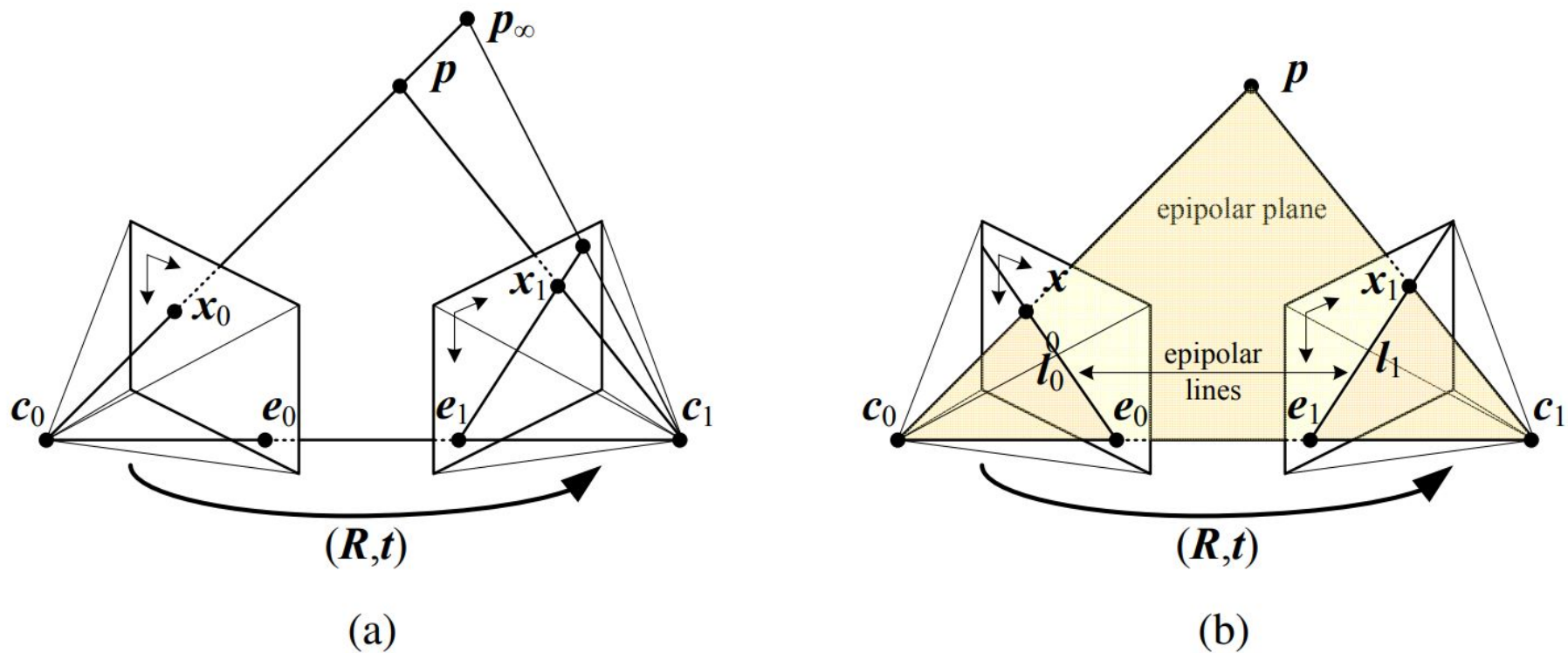








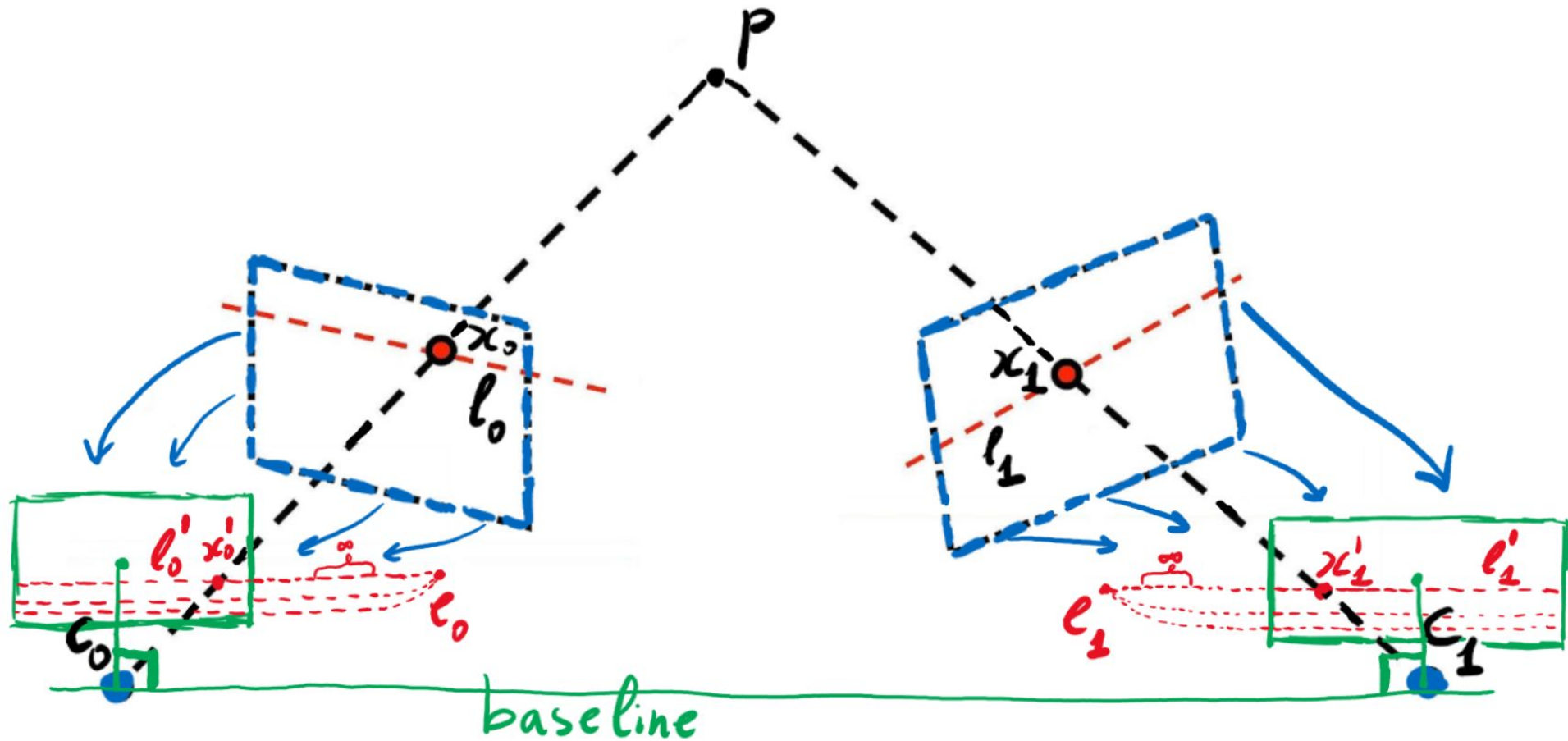
# Эпиполярная геометрия



**Figure 12.3** Epipolar geometry: (a) epipolar line segment corresponding to one ray; (b) corresponding set of epipolar lines and their epipolar plane.



# Ректификация: параллелим эпиполярные линии



Рассмотрим **воображаемую камеру** чья оптическая ось перпендикулярна прямой (**baseline**) между центрами камер. Иначе говоря выкинули эпиполисы на бесконечность (**epipole**: проекция одной камеры в другую - т.е. точка пересечения всех эпиполярных линий). Оставшаяся степень свободы - поворот вокруг оптической оси (**twist**) - делает эпиполярные линии горизонтальными.



# Ректификация

- 1) Поворачиваем так чтобы оптическая ось стала перпендикулярна прямой baseline, т.е. делаем эпиполярные линии параллельными.
- 2) Поворачиваем вокруг оптической оси (twist), т.е. делаем их горизонтальными.
- 3) Добавляем масштаб:
  - чтобы выровнять фокальные длины (если разные)
  - чтобы не терять информацию более детальной фотографии
  - чтобы ослабить горизонтальные искажения.

Image 1 and Epipole

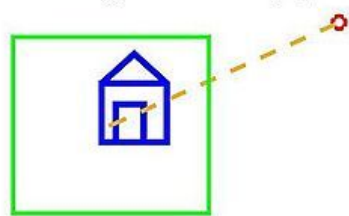


Image 2 and Epipole

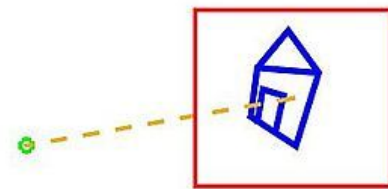


Image 1 Rotated

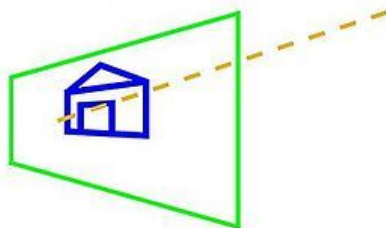


Image 2 Rotated

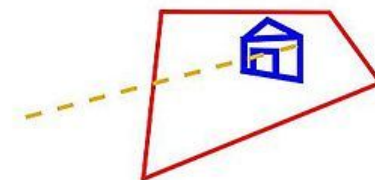


Image 1 Rotated and Twisted

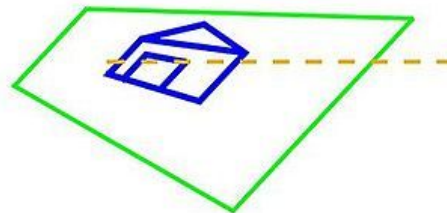


Image 2 Rotated and Twisted

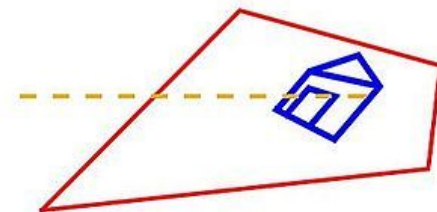


Image 1 Rectified

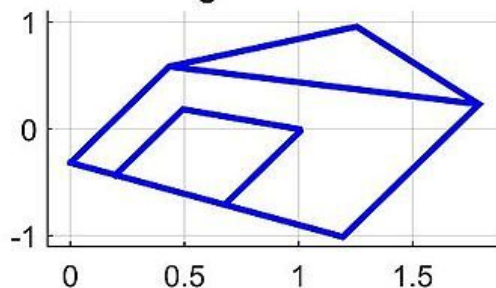
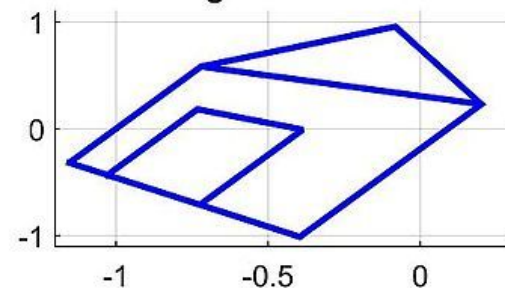
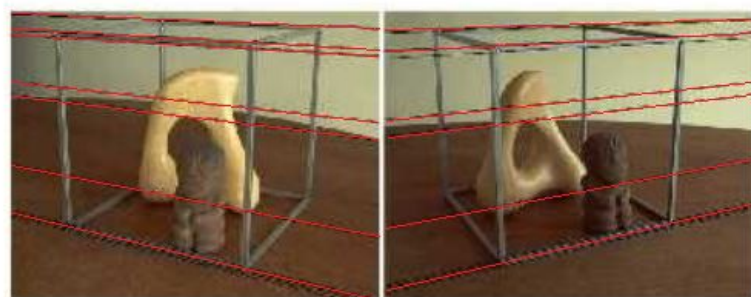


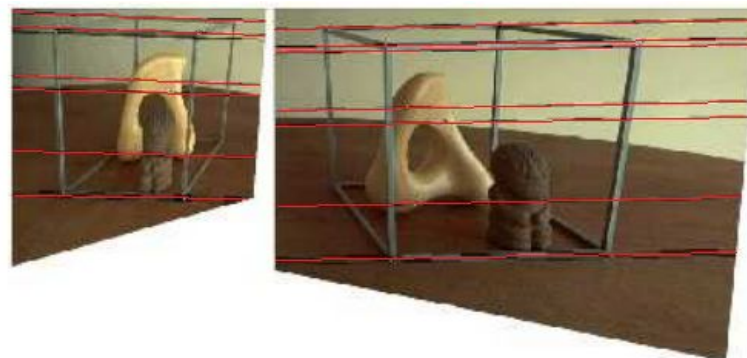
Image 2 Rectified



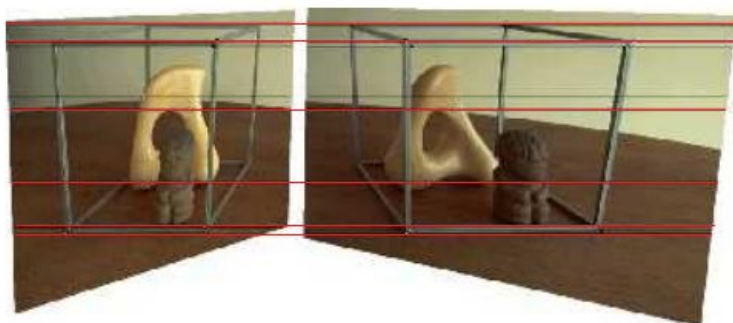
# Ректификация стереопары



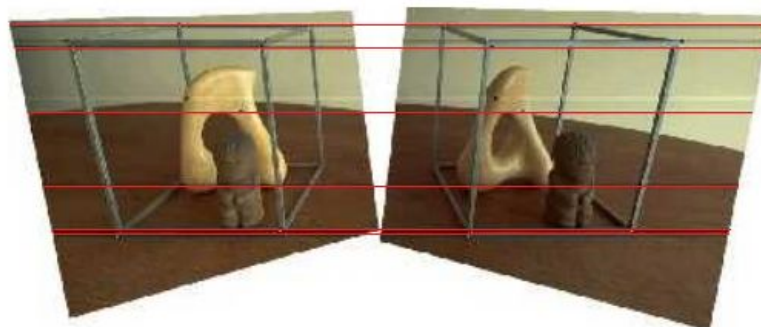
(a)



(b)



(c)



(d)

**Figure 12.4** *The multi-stage stereo rectification algorithm of Loop and Zhang (1999) © 1999 IEEE. (a) Original image pair overlaid with several epipolar lines; (b) images transformed so that epipolar lines are parallel; (c) images rectified so that epipolar lines are horizontal and in vertical correspondence; (d) final rectification that minimizes horizontal distortions.*



Fig. 11.11. **Image rectification.** (a) A pair of images of a house. (b) Resampled images computed from (a) using a projective transformation computed from  $F$ . Note, corresponding points in (b) match horizontally.

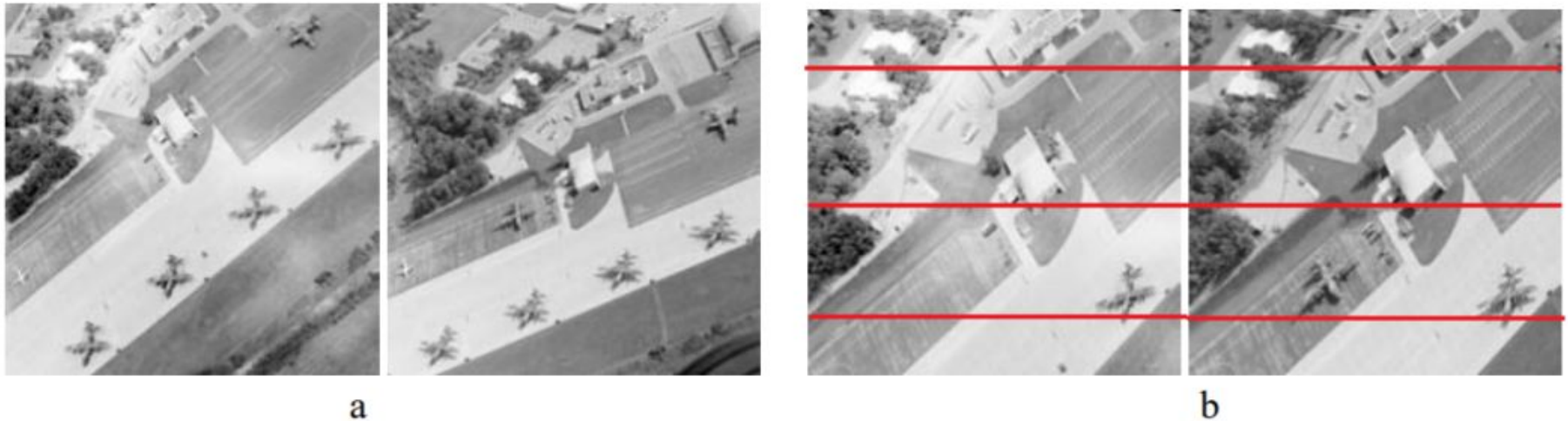
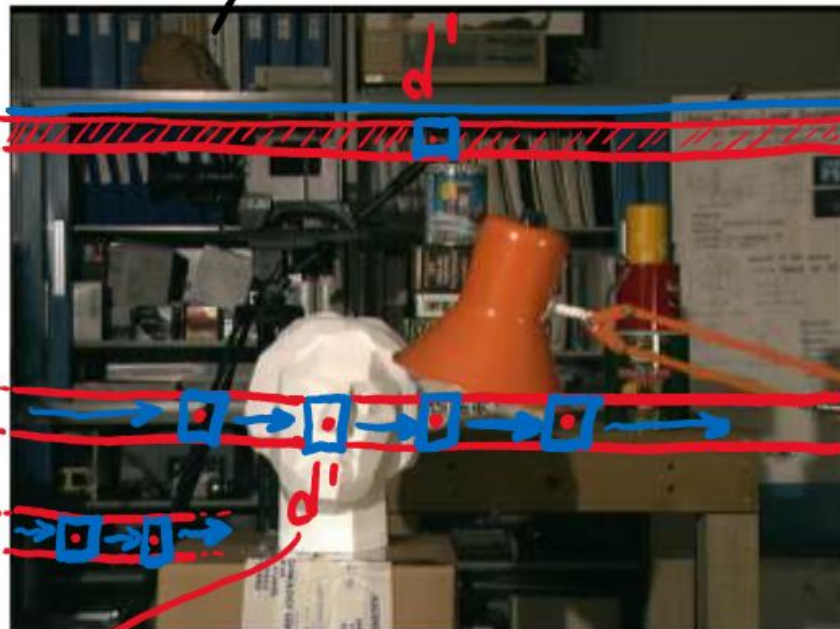
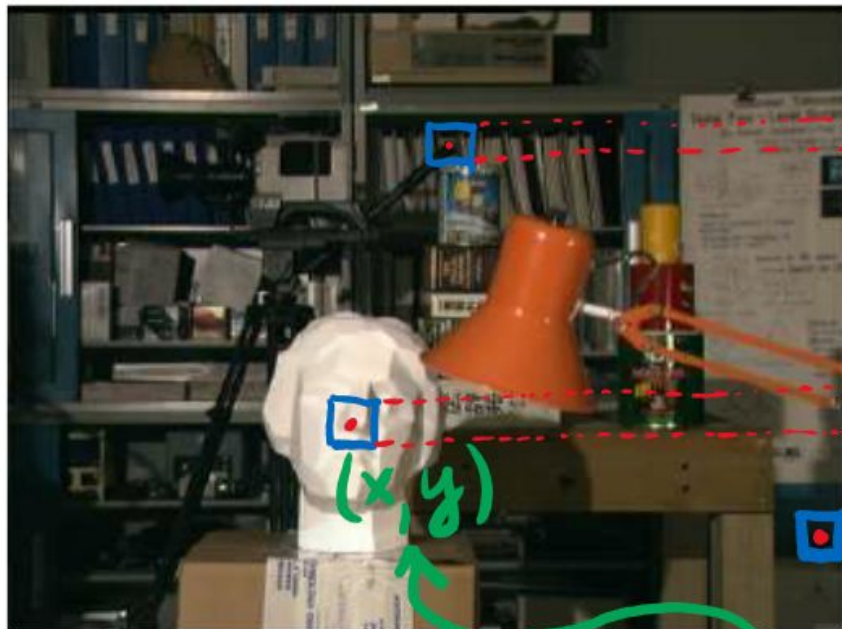


Fig. 11.12. **Image rectification using affinities.** (a) A pair of original images and (b) a detail of the images rectified using affine transformations. The average  $y$ -disparity after rectification is of the order of 3 pixels in a  $512 \times 512$  image. (For correctly rectified images the  $y$ -disparity should be zero.)



левая

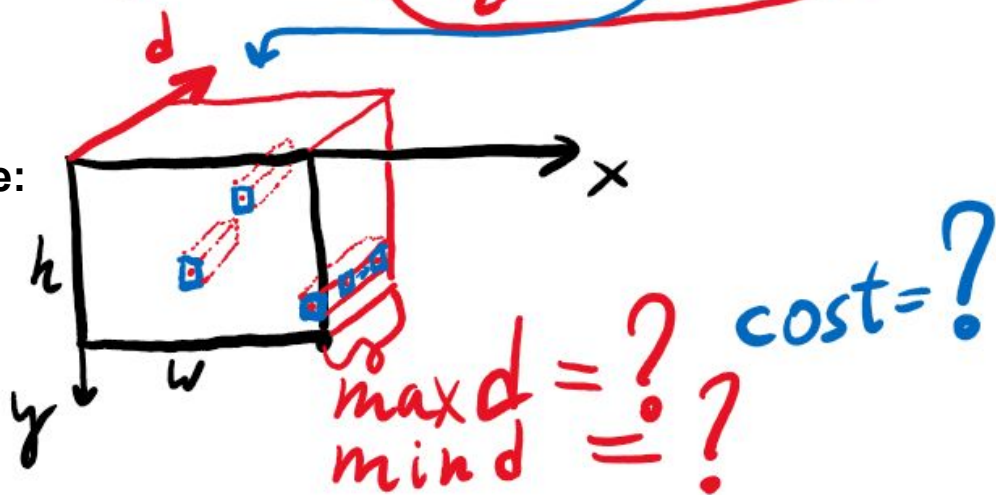
правая



$\rightarrow d$

$$d(x, y) = d' = \underset{d}{\operatorname{argmin}} \operatorname{cost}(x, y, d)$$

DSI (disparity space image) volume:





# Cost функции

1) Absolute Difference

$$C_{AD}(\mathbf{p}, \mathbf{d}) = |I_L(\mathbf{p}) - I_R(\mathbf{p} - \mathbf{d})|$$

2) Sum of Absolute Differences

$$C_{SAD}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |I_L(\mathbf{q}) - I_R(\mathbf{q} - \mathbf{d})|$$

Но что если яркость в картинках отличается?

# Cost функции

- 1) Absolute Difference
- 2) Sum of Absolute Differences

$$C_{AD}(\mathbf{p}, \mathbf{d}) = |I_L(\mathbf{p}) - I_R(\mathbf{p} - \mathbf{d})|$$

$$C_{SAD}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |I_L(\mathbf{q}) - I_R(\mathbf{q} - \mathbf{d})|$$

- 3) Zero-mean SAD

$$C_{ZSAD}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |I_L(\mathbf{q}) - \bar{I}_L(\mathbf{p}) - I_R(\mathbf{q} - \mathbf{d}) + \bar{I}_R(\mathbf{p} - \mathbf{d})|$$

$$\bar{I}(\mathbf{p}) = \frac{1}{N_{\mathbf{p}}} \sum_{\mathbf{q} \in N_{\mathbf{p}}} I(\mathbf{q})$$

# Cost функции

1) Absolute Difference

$$C_{AD}(\mathbf{p}, \mathbf{d}) = |I_L(\mathbf{p}) - I_R(\mathbf{p} - \mathbf{d})|$$

2) Sum of Absolute Differences

$$C_{SAD}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |I_L(\mathbf{q}) - I_R(\mathbf{q} - \mathbf{d})|$$

3) Zero-mean SAD

$$C_{ZSAD}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |I_L(\mathbf{q}) - \bar{I}_L(\mathbf{p}) - I_R(\mathbf{q} - \mathbf{d}) + \bar{I}_R(\mathbf{p} - \mathbf{d})|$$

$$\bar{I}(\mathbf{p}) = \frac{1}{N_{\mathbf{p}}} \sum_{\mathbf{q} \in N_{\mathbf{p}}} I(\mathbf{q})$$

4) Normalized Cross-Correlation

$$C_{NCC}(\mathbf{p}, \mathbf{d}) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} I_L(\mathbf{q}) I_R(\mathbf{q} - \mathbf{d})}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} I_L(\mathbf{q})^2 \sum_{\mathbf{q} \in N_{\mathbf{p}}} I_R(\mathbf{q} - \mathbf{d})^2}}$$

- Стат. оптимальна при гауссовом шуме

- Устойчива при разном значении gain (“масштаб” яркости) за счет деления на среднеквадратичное отклонение

- Чувствительна к выбросам (например на границе объекта, т.е. рядом с заслонением - с occlusion)

- И что если яркость в картинках отличается?



# Cost функции

1) Absolute Difference

$$C_{AD}(\mathbf{p}, \mathbf{d}) = |I_L(\mathbf{p}) - I_R(\mathbf{p} - \mathbf{d})|$$

2) Sum of Absolute Differences

$$C_{SAD}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |I_L(\mathbf{q}) - I_R(\mathbf{q} - \mathbf{d})|$$

3) Zero-mean SAD

$$C_{ZSAD}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |I_L(\mathbf{q}) - \bar{I}_L(\mathbf{p}) - I_R(\mathbf{q} - \mathbf{d}) + \bar{I}_R(\mathbf{p} - \mathbf{d})|$$

$$\bar{I}(\mathbf{p}) = \frac{1}{N_{\mathbf{p}}} \sum_{\mathbf{q} \in N_{\mathbf{p}}} I(\mathbf{q})$$

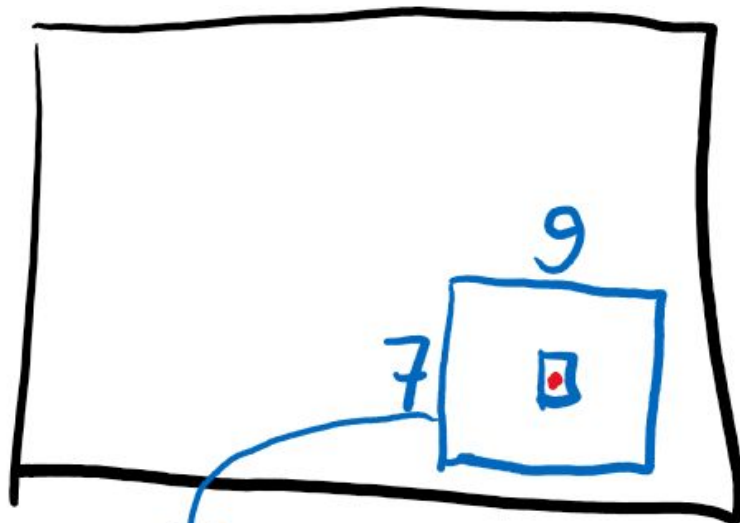
4) Normalized Cross-Correlation

$$C_{NCC}(\mathbf{p}, \mathbf{d}) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} I_L(\mathbf{q}) I_R(\mathbf{q} - \mathbf{d})}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} I_L(\mathbf{q})^2 \sum_{\mathbf{q} \in N_{\mathbf{p}}} I_R(\mathbf{q} - \mathbf{d})^2}}$$

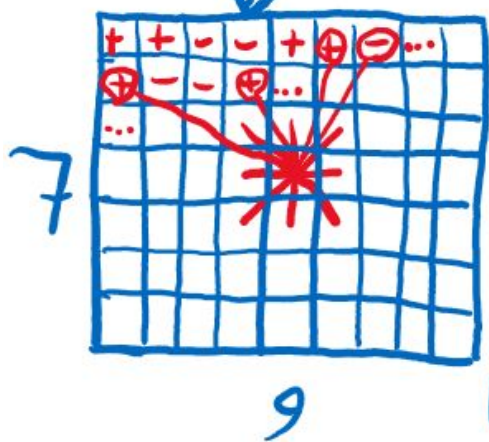
5) Zero-mean NCC

$$C_{ZNCC}(\mathbf{p}, \mathbf{d}) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_L(\mathbf{q}) - \bar{I}_L(\mathbf{p})) (I_R(\mathbf{q} - \mathbf{d}) - \bar{I}_R(\mathbf{p} - \mathbf{d}))}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_L(\mathbf{q}) - \bar{I}_L(\mathbf{p}))^2 \sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_R(\mathbf{q} - \mathbf{d}) - \bar{I}_R(\mathbf{p} - \mathbf{d}))^2}}$$

# Cost функции: Census



**Census** - битовый вектор описывающий патч вокруг пикселя и состоящий из  $7 \cdot 9 - 1 = 62$  бит его можно преподсчитать для обеих картинок стереопары.

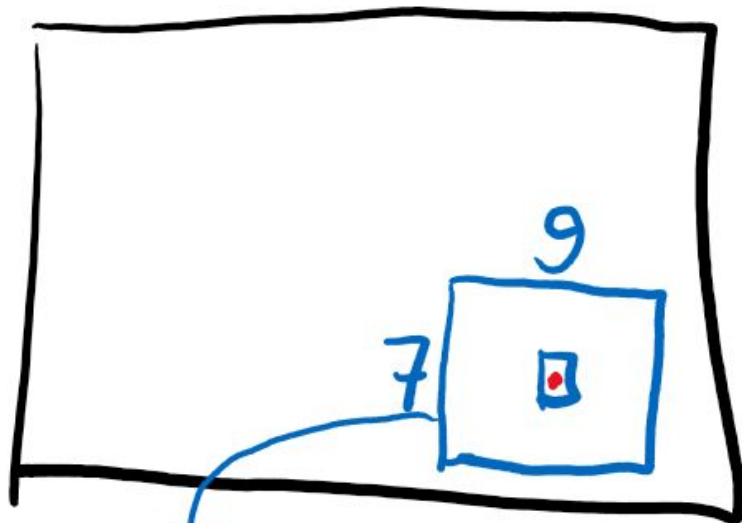


+  $\Leftrightarrow$  ярче центра  
-  $\Leftrightarrow$  темнее  
пикселе в центре

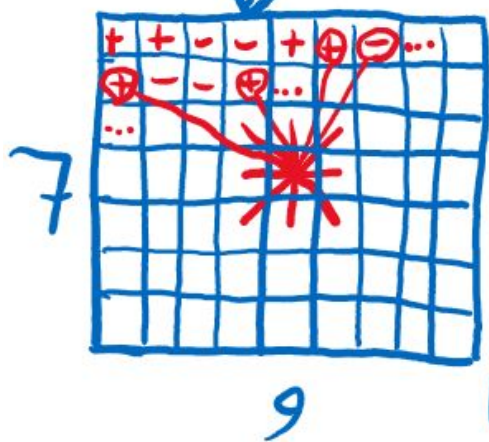
← patch[7][9]

Какая метрика похожести двух Census-ов?

# Cost функции: Census



**Census** - битовый вектор описывающий патч вокруг пикселя и состоящий из  $7 \cdot 9 - 1 = 62$  бит его можно преподсчитать для обеих картинок стереопары.



+  $\Leftrightarrow$  ярче центра  
-  $\Leftrightarrow$  темнее  
пикселе в центре

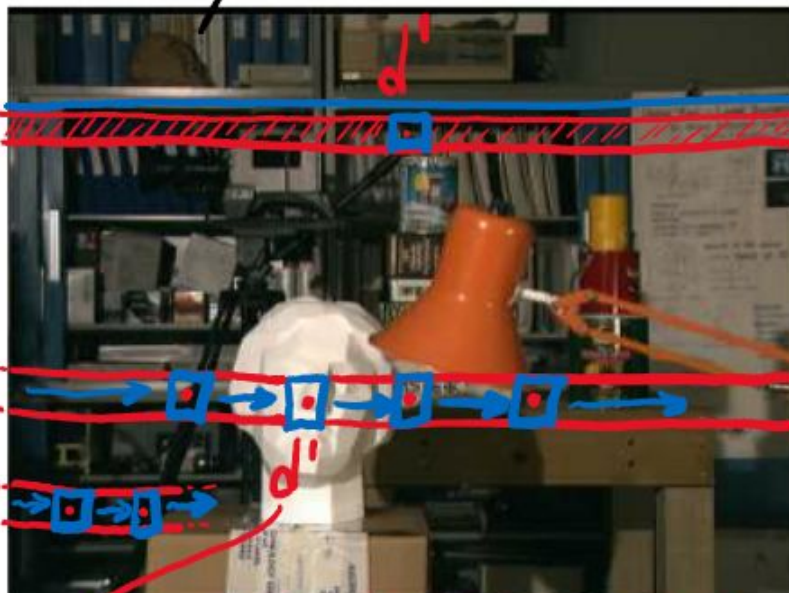
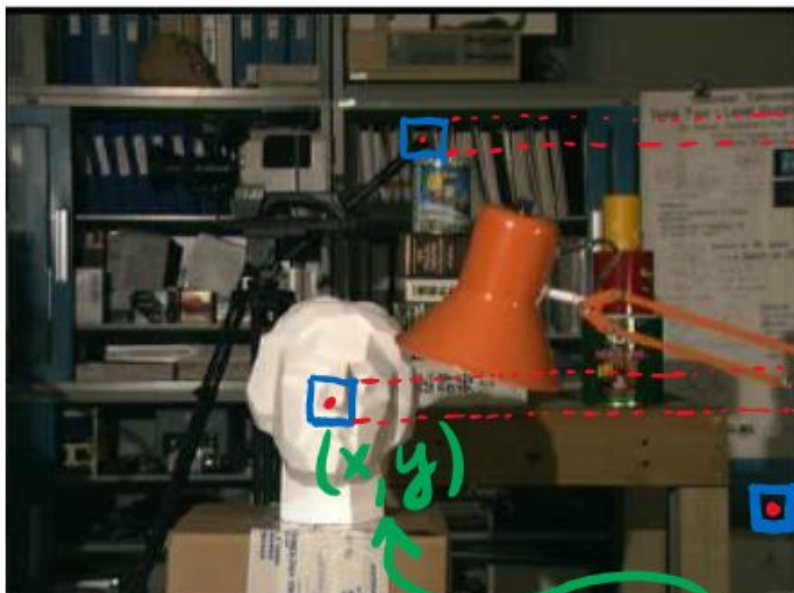
`patch[7][9]`

Метрика похожести двух Census:  
**Hamming distance**, т.е. просто  
**xor + popcount!**



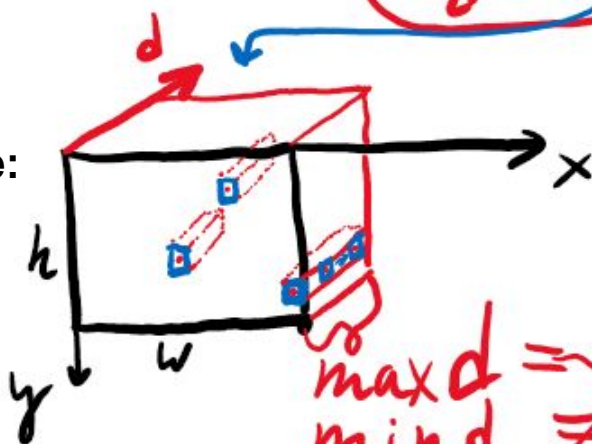
левая

правая



$$d(x, y) = d' = \underset{d}{\operatorname{argmin}} \operatorname{cost}(x, y, d)$$

DSI (disparity space image) volume:



cost = 😊

кажется  
одну из 3D  
камерных морек

Как добиться субпиксельной точности?

## Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- **Parabola fitting**

## Как профильтровать ошибки и occlusions (заслоненности)?

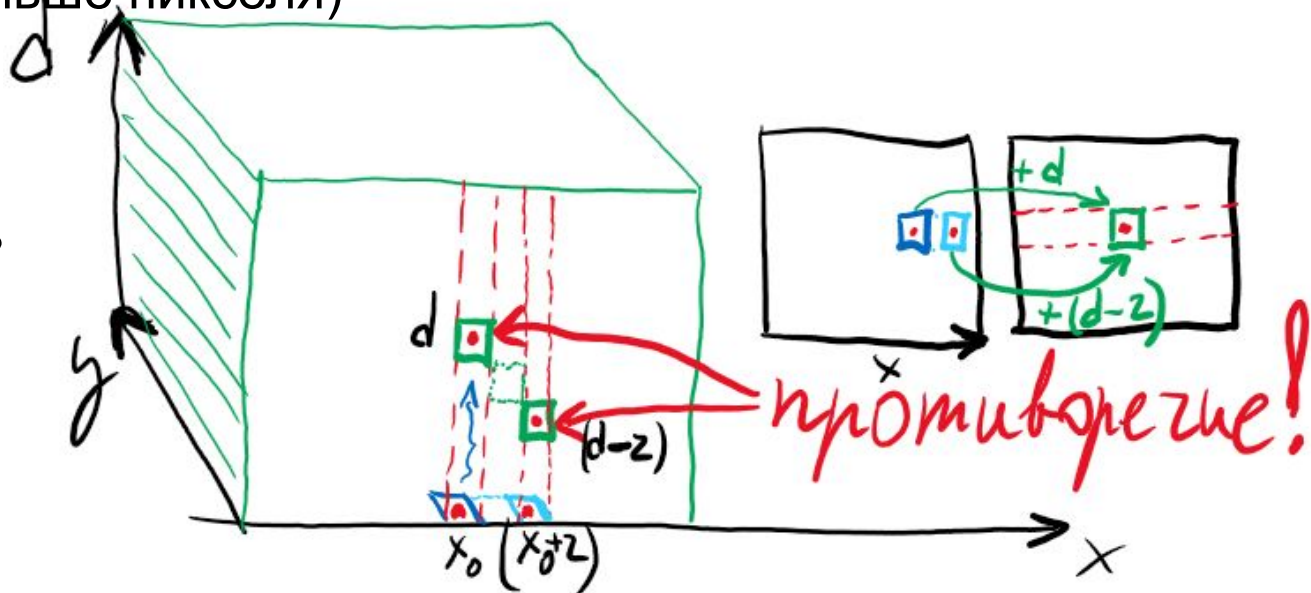


# Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- **Parabola fitting**

## Как пофильтровать ошибки и occlusions (заслоненности)?

- Проверяем **one-to-one** mapping через **left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаратета меньше пикселя)
- Проверяем **one-to-one** mapping через диагональ в **DSI volume**:



## Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- **Parabola fitting**

## Как пофилтровать ошибки и occlusions (заслоненности)?

- Проверяем **one-to-one** mapping через **left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаратета меньше пикселя)
- Проверяем **one-to-one** mapping через диагональ в **DSI volume**

## Как справляться с неоднозначностями, регулярными и слаботекстурированными поверхностями?

## Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- **Parabola fitting**

## Как пофилтровать ошибки и occlusions (заслоненности)?

- Проверяем **one-to-one** mapping через **left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаратета меньше пикселя)
- Проверяем **one-to-one** mapping через диагональ в **DSI volume**

## Как справляться с неоднозначностями, регулярными и слаботекстурированными поверхностями?

Глобальной оптимизацией:

- поощряют совпадающие и близкие disparity значения (чтобы тяготеть к связным поверхностям)
- штрафовать разрывы (чтобы их не было слишком много)



# Глобальная оптимизация

Минимизируем энергию:

$$E(d) = E_D(d) + \lambda E_S(d)$$

Где **data term**:  $E_D(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_S(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где  $\rho$ - монотонная функция штрафа

Как ослабить штраф за разрыв на границе объекта?

# Глобальная оптимизация

Минимизируем энергию:

$$E(d) = E_D(d) + \lambda E_S(d)$$

Где **data term**:  $E_D(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_S(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где  $\rho$ - монотонная функция штрафа

Как ослабить штраф за разрыв на границе объекта?

Домножать с учетом перепада яркости (или силы градиента):

$$\rho_D(d(x, y) - d(x + 1, y)) \cdot \rho_I(\|I(x, y) - I(x + 1, y)\|)$$

# Глобальная оптимизация

Минимизируем энергию:

$$E(d) = E_D(d) + \lambda E_S(d)$$

Где **data term**:  $E_D(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_S(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где  $\rho$ - монотонная функция штрафа

Это NP-полная задача, приближенное решение можно искать с помощью разных методов, например через Markov Random Fields или минимальный разрез графа. Но все это очень медленно работает!

# Глобальная оптимизация

Минимизируем энергию:

$$E(d) = E_D(d) + \lambda E_S(d)$$

Где **data term**:  $E_D(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_S(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где  $\rho$ - монотонная функция штрафа

Это NP-полная задача, приближенное решение можно искать с помощью разных методов, например через Markov Random Fields или минимальный разрез графа. **Но все это очень медленно работает!**



# Semi-Global Matching (SGM)

$$E(D) = \sum_{\mathbf{p}} \left[ C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right]$$

все пиксели  $\rightarrow \mathbf{p}$   
 локальные costs  
 соседние пиксели  $\mathbf{p}$   
 штраф за отличие disparity на 1  
 штраф за разрыв (отличие  $\geq 1$ )

$$P_2 \geq P_1$$

можно задать последнее на границах  $P_2 = \frac{P_2'}{|I_{b\mathbf{p}} - I_{b\mathbf{q}}|}$

# Semi-Global Matching (SGM)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению  $\mathbf{r}$ :

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$$

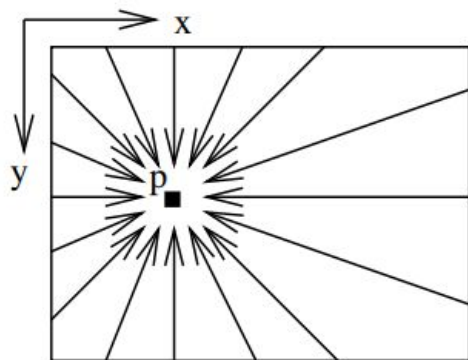
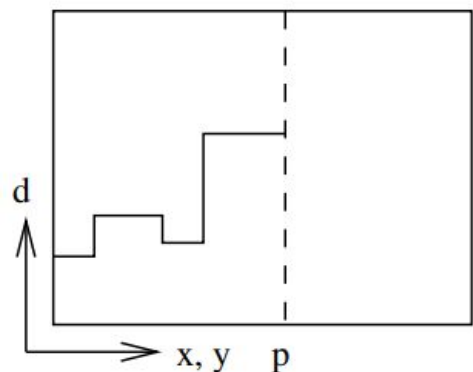
*cost* (blue arrow pointing to  $C(\mathbf{p}, d)$ )

*направление* (red arrow pointing to  $\mathbf{r}$ )

*пиксель шаг  $\mathbf{r}$  назад* (red text below the  $k$  term)

(a) Minimum Cost Path  $L_{\mathbf{r}}(\mathbf{p}, d)$

(b) 16 Paths from all Directions  $\mathbf{r}$



# Semi-Global Matching (SGM)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

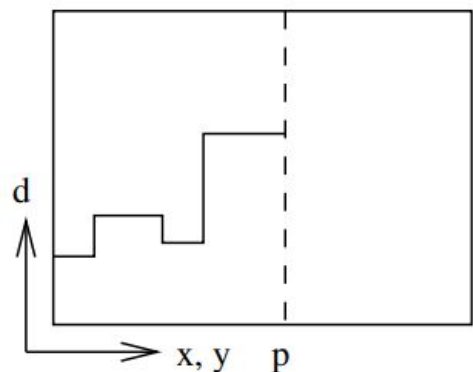
Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению  $\mathbf{r}$ :

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$$

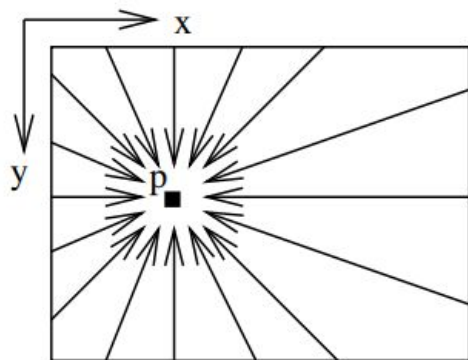
*Handwritten annotations:*

- $L_{\mathbf{r}}(\mathbf{p}, d)$ : *направление* (direction)
- $C(\mathbf{p}, d)$ : *cost* (cost)
- $L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d)$ : *то же смещение* (the same displacement)
- $\min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2$ : *пиксель шаг r назад* (pixel step r back)

(a) Minimum Cost Path  $L_{\mathbf{r}}(\mathbf{p}, d)$



(b) 16 Paths from all Directions  $\mathbf{r}$



# Semi-Global Matching (SGM)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению  $\mathbf{r}$ :

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d),$$

$$L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1,$$

$$\min L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) + \min L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$$

*cost* (blue arrow pointing to  $C(\mathbf{p}, d)$ )

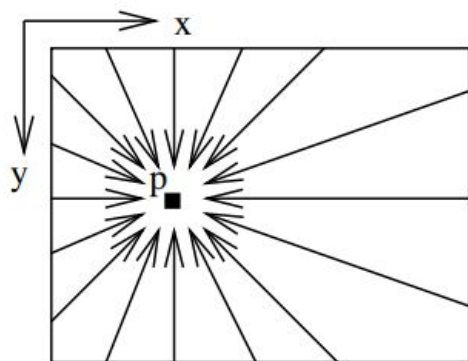
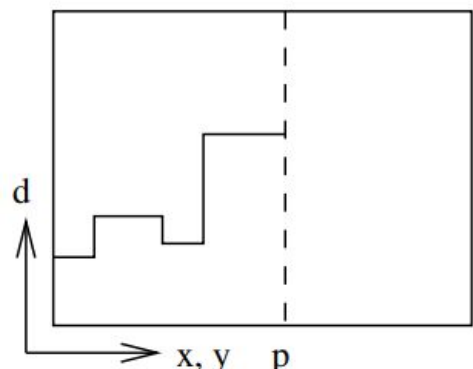
*направление* (red arrow pointing to  $\mathbf{r}$ )

*соседнее смещение* (green arrow pointing to  $L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1$ )

*пиксель шаг  $\mathbf{r}$  назад* (red arrow pointing to  $\mathbf{p} - \mathbf{r}$ )

(a) Minimum Cost Path  $L_{\mathbf{r}}(\mathbf{p}, d)$

(b) 16 Paths from all Directions  $\mathbf{r}$





# Semi-Global Matching (SGM)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению  $\mathbf{r}$ :

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \min_{i} L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$$

*cost* (blue arrow pointing to  $C(\mathbf{p}, d)$ )

*направление* (red arrow pointing to  $\mathbf{r}$ )

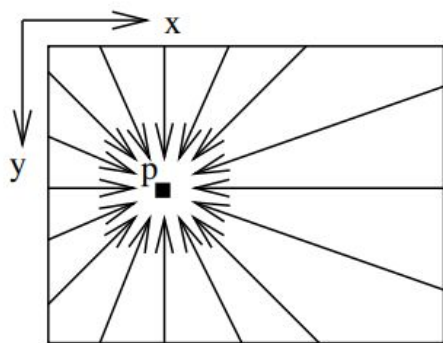
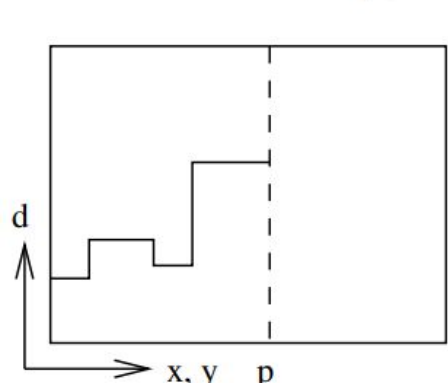
*разрыв* (green oval around the  $\min_i$  term)

*против пиксель шаг r назад* (green arrows pointing to  $\mathbf{p} - \mathbf{r}$  and  $k$ )

*перекос* (green arrow pointing to  $k$ )

(a) Minimum Cost Path  $L_{\mathbf{r}}(\mathbf{p}, d)$

(b) 16 Paths from all Directions  $\mathbf{r}$



# Semi-Global Matching (SGM)

- 1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
- 2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
- 3) **Matching cost computation**: посчитали cost во всем DSI volume
- 4) **Cost support aggregation**: прошли 16-ю волнами - оценили энергию во всех гипотезах всего объема
- 5) **Winner takes all (WTA)**: для каждого пикселя нашли disparity с минимальной энергией - это победитель
- 6) **Refinement**: субпиксельно уточнили disparity-победителя с учетом энергии соседей
- 7) **Left-right check** или проверка **one-to-one mapping** по диагонали ради удаления ошибок и регионов где не видит один из кадров (**occlusion**)

# Semi-Global Matching (SGM)

- 1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
- 2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
- 3) **Matching cost computation**: посчитали cost во всем DSI volume
- 4) **Cost support aggregation**: прошли 16-ю волнами - оценили энергию во всех гипотезах всего объема
- 5) **Winner takes all (WTA)**: для каждого пикселя нашли disparity с минимальной энергией - это победитель
- 6) **Refinement**: субпиксельно уточнили disparity-победителя с учетом энергии соседей
- 7) **Left-right check** или проверка **one-to-one mapping** по диагонали ради удаления ошибок и регионов где не видит один из кадров (**occlusion**)

Но  $O(W*H*D)$  памяти! (и времени)

# SURE tSGM

Воспользуемся **coarse-to-fine** схемой!

(т.е. будем прогрессировать по пирамиде детализаций стереопары, постепенно уточняя карту диспаритета)

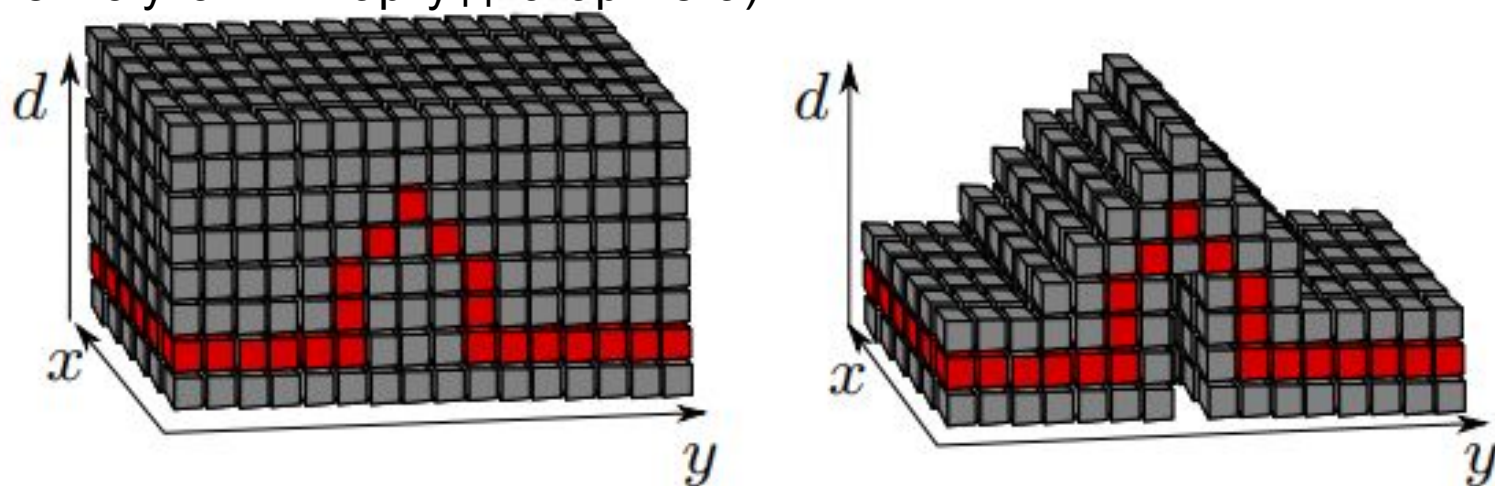


Figure 2: Cost structures of classic SGM (left) and tSGM (right). Red cubes represent costs for the true correspondences. Gray cubes mark the costs of potential correspondences, thus the disparity search ranges.



# SURE tSGM

Как при переходе от менее детальной стереопары и карты глубины к более детальной решить в каком диапазоне требуется выполнять поиск?  
(т.е. агрегацию энергии по 16 направлениям)

# SURE tSGM

После определения карты диспаратета очередного уровня - по ней определяем диапазоны ее пикселей:

- 1) Если пиксель был успешно сопоставлен - его диапазон это минимум и максимум по диспаратетам в окне  $7 \times 7$ .
- 2) Если пиксель не был сопоставлен - минимум и максимум по диспаратетам в окне  $31 \times 31$ , а в сам пиксель кладем медиану диспаратета по этому же окну  $31 \times 31$ .

Ограничиваем диапазон до 16 и 32 соответственно.

И наконец устанавливаем **диапазон** для следующей ступени детализации:

$$\left[ 2 * (x_b + d - d_{min}), 2 * (x_b + d + d_{max}) \right]$$

Итого чаще всего высота столбика - 32 ячейки.

На границах или в сложных местах - 64.

# SURE tSGM

С памятью и скоростью стало гораздо лучше. Теперь  $O(W*H*64)$ .

Как найти **первый уровень?**

# SURE tSGM

С памятью и скоростью стало гораздо лучше. Теперь  $O(W*H*64)$ .

Как найти **первый уровень**?

- Построить экстраполировав смещения сопоставленных ключевых точек
- Либо просто построить с диапазоном поиска размера  $64 >$  ширина самого недетального первого уровня

Можно ли реализовать **left-right check** без двукратного замедления?



# SURE tSGM

С памятью и скоростью стало гораздо лучше. Теперь  $O(W*H*64)$ .

Как найти **первый уровень**?

- Построить экстраполировав смещения сопоставленных ключевых точек
- Либо просто построить с диапазоном поиска размера  $32 >$  ширина самого недетального первого уровня

Можно ли реализовать **left-right check** без двукратного замедления?

- Строим обе карты глубины для всех уровней кроме самого детального
- Left-right check сверку для левой камеры делаем с  $x2$  менее детальной картой смещений правой камеры (увеличив допуск суммарного смещения с  $<1.0$  до  $<2.0$ )

# SURE tSGM

- 1) Что делать с тонкими структурами?  
(усы котика, перила у лестницы, антенны)
- 2) Что делать с бликами и регулярными паттернами?



# SURE tSGM

- 1) Что делать с тонкими структурами?  
(усы котика, перила у лестницы, антенны)
- 2) Что делать с бликами и регулярными паттернами?



# SURE tSGM

- 1) Что делать с тонкими структурами?  
(усы котика, перила у лестницы, антенны)
- 2) Что делать с бликами и регулярными паттернами?





# Ректификация, SGM, tSGM. Ссылки

Книга (про ректификацию, SGM):

- [Computer Vision: Algorithms and Applications, Richard Szeliski](#)

Cost functions:

- [Evaluation of Stereo Matching Costs on Images with Radiometric Differences, Hirschmuller, 2008](#)

SGM:

- [Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information, Hirschmuller, 2005](#)
- [Stereo Processing by Semi-Global Matching and Mutual Information, Hirschmuller, 2008](#)

SURE tSGM:

- [SURE: PHOTOGRAMMETRIC SURFACE RECONSTRUCTION FROM IMAGERY, Rothermel et. al., 2013](#)

# Вопросы?



Полярный Николай  
[polarnick239@gmail.com](mailto:polarnick239@gmail.com)